"Express Mail" mailing label number EL 003481995 US
Date of Deposit OCTOBER 20, 1998
I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" services under 37 C.F.R. 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D.C. 20231.
Typed Name of Person Mailing Paper or Fee: MAGGIE DEEDS
Signature: Maggie Deeds

UTILITY PATENT APPLICATION

NETWORK DOCUMENT DELIVERY SYSTEM

INVENTORS:

Steven C. Johnson, Eagle, Idaho; Dennis W. Hicks, Eagle, Idaho; Richard Newman, Meridian, Idaho; Gary Johnson, Boise, Idaho; Lisa O'Toole, Eagle, Idaho; David Hay, Boise, Idaho; Chris Gyllenskog, Boise, Idaho; Matt Stephenson, Stratford, Wisconsin; Frank Hartmann, Boise, Idaho; Ray Asbury, Boise, Idaho; Eric Luttman, Boise, Idaho.

TITLE OF INVENTION: Network Document Delivery System

INVENTORS: Steven C. Johnson, Eagle, Idaho; Dennis W. Hicks, Eagle, Idaho; Richard D. Newman, Meridian, Idaho; Gary Johnson, Boise, Idaho; Lisa O'Toole, Eagle, Idaho; David Hay, Boise, Idaho; Chris Gyllenskog, Boise, Idaho; Matt Stephenson, Stratford, Wisconsin; Frank Hartmann, Boise, Idaho; Ray Asbury, Boise, Idaho; Eric Luttmann, Boise, Idaho.

Microfiche Appendix: This specification has appended hereto a microfiche appendix having 17 fiche containing approximately 4600 frames.

RELATED APPLICATIONS: This application claims the priority of Provisional Application Serial No. 60/063,891 filed October 22, 1997.

DESCRIPTION

BACKGROUND OF THE INVENTION

Technical Field. This invention generally relates to document handling on a computer network, and more particularly, this invention relates to a recipient based system for printing, faxing, storing and transmitting electronic documents across networks.

Sub A17

Background. Modern business requires that computing environments become more flexible, easy to use, allow for growth, and in particular, be measurably cost effective. A fundamental element of computing environments is the handling of documents. The concept of a "document" is now much more than just a printed piece of paper. A document can be printed in both black and white and color, it can be viewed electronically, it can be archived on removable or fixed storage media, and it can be transmitted electronically. Unfortunately, the traditional mechanisms for delivering documents consist of independent solutions. This problem is characteristic of the current device based paradigm for document delivery. It would therefor be desirable to provide a single integrated solution which allows a network

user to deliver his or her document to one or more different destinations or recipients in a single step regardless of the end form in which the document is presented.

SUMMARY OF THE INVENTION

According to the present invention, this general end is achieved by a system of networked computers, peripherals and document delivery software which provides the user with a familiar simple user interface, such as a print dialog box in a Windows® environment, to deliver documents to a variety of different destinations, both within the network, across networks and outside of the network via remote links.

In one embodiment of the invention, a document generation device participating in the system, whether directly connected to the network or interacting full or part time through a remote link, may be provided with a print driver which translates an electronic document into a non-specific, or printer independent, printer language file and appends to this file a job ticket containing any other rendering characteristics which may not be supported by the printer independent language. Rendering characteristics include such things as color or monochrome output, duplex printing, number of original copies, stapling, collating, binding, recipient and destination information, etc. This entire file is then transmitted to the system server which analyzes the file, including the rendering characteristics; determines the best output device(s); appends output device specific commands to the general printer language file; and transmits this file to the device(s).

The job ticket and related flexibility of the software also enable recipient based delivery and result based delivery, both of which represent a paradigm shift away from device based printing. Recipient based delivery focuses upon the location of a particular recipient and the medium through which that recipient prefers to receive information, as opposed to a particular printer in the general location of the recipient. Result based delivery focuses on the presentation and medium for delivery of information, as opposed to a particular device or device location.

In one embodiment of the invention, the software on the server assigns an affinity value to each print job based upon the job size, destination and rendering characteristics. This affinity value is then used to determine which output

device(s) will receive the document. The server must therefor be aware of what output devices are participating in the system, where they are located, what their specific characteristics are and whether or not any particular device is currently available. This information may be gathered automatically by having the server poll for network resources, the information may be manually entered by a user or system administrator, or the information may be input by a combination of the two methods. The user may elect to bypass the invention by selecting a specific printer driver rather than that of the invention. In this case, the invention software on the server simply forwards the print job on to the specific printer requested.

This system facilitates the ability to implement many other valuable and desirable features. One such feature is the ability to distribute a large job over two or more output devices participating in the system, essentially defining multiple output devices as a single output device. This is most advantageous where a single job contains multiple original copies and each output device receives one or more entire copies to output, thereby decreasing output time by a factor of the number of output devices and not causing the user to collate pages from multiple output devices. Additionally, the invention can distribute jobs over the output resources on the network to even distribute the workload.

Another feature which may be implemented is an activity log or journal which can provide detailed information concerning usage. The log can provide such information as the size and number of print jobs requested by any combination of users for billing purposes; job completion verification; diagnostic information to allow an operator to determine when and why jobs failed; and resource utilization information such as toner usage for a printer to plan for inventory, expenses and maintenance. The journal may be kept in a standard database format which may be easily imported to accounting, database or other computer applications.

The invention can support virtually any output device such as: standard image forming devices including printers, plotters, and video; facsimile devices; email communications; data communications links; and archival devices. In the case of hard copy image forming devices such as a laser printer, both banner and receipt pages can be generated. Banner pages can be used to identify sets of

jobs on each printer and notify the operator of any finishing operations to be performed. Receipt pages can be used to provide a short job summary and verify job completion. Supported data communications can include serial telecommunications via data modems, network communications using TCP/IP, NetBEUI, IPX/SPX and ETHERTALK. Supported storage devices for archival purposes as well as job submission, include floppy diskettes, IOMEGA JAZ drives and SYQUEST SYJET drives.

Advantageously, archival and storage of documents may be done in a platform independent format such as ADOBE's Portable Document Format (PDF). PDF allows a user of virtually any operating system to view and print archived documents using a freely distributed viewing program, ADOBE ACROBAT READER.

In another embodiment of the invention, the software on the server can be configured so that a job sent to a specific port or by a particular type of printer driver is always output the same way or according to a specific set of rules. This enables document generation devices to use a standard printer specific printer driver, such as a HEWLETT PACKARD LASERJET driver, and still have the job output to one or more different devices.

Additional advantages and novel features of the invention will be set forth in part in the description that follows, in the attached appendix and in part will become apparent to those skilled in the art upon examination of the following or may be learned by practice of the invention. The objects and advantages of the invention may be realized and attained by means of the instrumentalities and combinations particularly pointed out in the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a schematic representation of a network document delivery system according to the invention;

Fig. 2 is a block diagram illustrating the functional aspects of the software;

Fig. 3 is a graphic representation of the connections that may be made to a representative hardware system;

Fig. 4 is a graphic representation of a high level system object model of the invention;

Fig. 5 is a graphic representation of a subordinate level object model showing the relationship between a task and a job;

Fig. 6 is a graphic representation of a subordinate level object model showing the relationship between users and objects;

Fig. 7 is a graphic representation of a subordinate level object model showing the relationship between the general product and different types of output;

Fig 8 is a graphic representation of a subordinate level object model showing the relationship between a data source and the data port;

Fig. 9 is a graphic representation of a subordinate level object model showing the relationship between a general device, a pool of devices, an atomic device, a remote system, a system device and an array of devices;

Fig. 10 is a graphic representation of a subordinate level object model showing the hierarchical relationship between a system device and an atomic device;

Figs. 11 and 12 describe the general life cycle model and rely on the schemata of Figs. 13 through 55;

Fig. 56 describes the Fusion notation used in Figs. 4 through 10;

Fig. 57 describes the Life Cycle Model notation used in Figs. 11 and 12;

Figs. 58 through 61 describe various notation schemes used in Figs. 62 through 66;

Fig. 62 is an object interaction graph illustrating how affinity is determined;

Fig. 63 is an object interaction graph illustrating the submit job sequence;

Fig. 64 is an object interaction graph illustrating the instruct to job ticket sequence;

Fig. 65 is an object interaction graph illustrating the instruct to instruction set sequence;

Fig. 66 is an object interaction graph illustrating the normal execute sequence; and

Figs. 67 through 81 illustrate one possible graphical user interface for the job ticket and show some of the various delivery options.

DETAILED DESCRIPTION OF THE INVENTION

Referring now to the figures, one possible network document delivery system, generally designated at 10 in the figures, according to the invention will be described in detail. The network shown includes at least one document generator 11, such as a networked personal computer, having a client user interface 12 installed therein; a server 13 having main job processing software 14 therein including a server user interface 15; and two or more document output devices 16. Fig. 3 shows a representative hardware connection configuration or network on which the invention may be implemented.

The simplified user model illustrated in Fig. 2 provides a procedural view of system operation. In this model, the overall system may include a main program 14, multiple data sources, such as client print driver 17, and/or other input clients, such as a manufacturer specific print driver 18, and multiple output devices 16.

A job is sent from a data source such as document generator 11 to main program 14 via a data port 19. A job must contain a data stream to be rendered, also referred to as image data, on some output device. The job may also contain a job ticket, which is a collection of specific information concerning the desired output presentation, such as a standard hard copy print job; a fax; an archival; an email; finishing features; routing information; and even billing information.

In the case, where print driver 17 is used, here when the user selects "auto" as the print destination, job ticket information is provided by client print driver 17. In this case print driver 17 includes a generic language translator 24 which translates an electronic document into a non-specific, or printer independent, printer language file and appends to this file a job ticket containing any other rendering

characteristics which may not be supported by the printer independent language. In other cases, the job information may be provided by 'default' job tickets or port profiles associated with a data port, a user name which can be determined from the network name, or a system default job ticket.

Job parser 20 examines the incoming job for a job ticket and applies default job tickets as required, then sends the job to routing and affinity process 21. Routing and affinity process 21 determines the capabilities required to complete the job successfully and the affinity of each potential output device for the job. Routing and affinity process 21 assigns an affinity value to each print job based upon the job size, destination and rendering characteristics by comparing the requested features with the available features logged in resource library 25. Available resources may be gathered and logged into resource library 25 by server 13 automatically by polling the network for resources. Additionally, the information may be manually entered by a user or system administrator or it may be input by a combination of the two methods. The job is then routed to a device specific assembler 22, also sometimes called the 'transform', to change the image data to a device specific form. The image data is then sent to the appropriate output device(s) 16 via a communications channel 23. In addition, the current status of each device can be monitored by the main program via communication channel 23.

Most commonly, output devices 16 are printers, but they can also be fax machines, electronic storage media, such as a 'file' on diskette, removable media, hard disk, tape drive, network drive, etc., or even email.

The simplified model can be extended to include multiple data ports with an associated default job ticket or port profile for each. A combination of port and port profile is referred to as a 'virtual queue'. Also, note that client print driver 17 can reside on the same host as main program 14, so that the operator of main program 14 can also submit jobs.

While the simplified user model illustrated in Fig. 2 provides a procedural view of system operation, the following illustrative embodiment takes advantage of the multitasking nature of a host operating system, such as WINDOWS

NT and the capabilities of object oriented programming techniques. This embodiment is illustrated in Fig. 4.

Here, the main program is actually a set of programs running simultaneously. Also, the job parsing, routing, and assembling functions are spread out over a set of objects. One possible set of object models are shown in Figs. 4 - 10. An explanation of the notation is included in Figs. 56 - 61, but in general, diamonds show relationships between models, and triangles denote hierarchy. For instance, referring to Fig. 5, a job is a task, and a job includes at least one document. Tasks may have one or more parent/child relationships. External agents, such as the human operator, are also represented by objects, even though the object may not have a corresponding software implementation.

Each of the objects shown in Fig. 4 can be decomposed or broken down into other objects as shown in the other figures. The objects enclosed within dashed lines are programs. The Operator, Recipient, and User are people as shown in Fig. 6. A Product is the output of the system as shown Fig. 7.

A job is created by a data source such as a document generator 11 and more specifically, usually by client print driver 17. Fig. 8 shows a more detailed view of possible types of data sources and their relationships to data ports. Note that remote systems can send jobs just like any local source. Likewise, a remote system may be configured as a device. This allows passing of a job from system to system, in a distributed network-like manner. The purpose for configuring the systems this way is to reduce phone charges by using local area network (LAN) communications between main systems. This allows jobs to be passed to LAN or phone connected printers, even though the printers are not available to the local system.

The data source passes the job to the data port. Note that everything inside the area surrounded by the dotted line, including the data port, are the main programs. The job parsing function is performed by the data port. The port creates a job object in the system that includes a document, i.e. image data, and job ticket as shown in Fig. 5. The job ticket may need to be formed from an associated port and/or user profile, i.e. a default job ticket. The job ticket is designed to allow routing

of the job to the best device and storing of data for billing and management purposes. The job ticket allows separation of the job specific features, such as number of copies, finishing, recipient information, etc., from the image data. Eventually job specific information needs to be in a form unique to each printer or output device, depending on its manufacturer and its configuration as it was installed, as some finishing features such as sorters and staplers are optional. The specialization of the generic or device independent data stream to the actual production device data stream is done after the production device is chosen by the system.

The data port creates a job and passes it to the system device. Fig. 9 shows the device hierarchy, and Fig. 10 shows the device relationships. There is only one system device in the system, and it is always the first device to receive each job. Every device examines the job to see if it can produce it, decompose it into tasks, or route it to a child device. Thus, the devices contain the routing function shown in the simplified model. All devices are implemented as objects. Device objects are serialized meaning configuration parameters are stored to disk so that they may be restored after a system reset.

A key feature of the device design is the relationship between pools, arrays, and atomic devices as illustrated in Fig. 10. A pool is generally a grouping of like devices. The grouping can be by function such as faxes, printers, or archive devices, or by some other criteria such as location, e.g. all printers on the second floor, permissions, or routing. An array is a collection of like devices. An atomic device represents the smallest whole constituent part. As far as the parent pool is concerned, an array is an atomic device, and thus the array class is derived from the atomic device class. At the lowest level, an atomic device 'knows' that it is capable of producing a product, and thus will determine its capabilities and will calculate an affinity for a given job. All devices are ultimately derived from a single device class. This design pushes specialization to the lowest possible level. If a device needs a particular resource to produce a product, fonts or electronic forms for example, it submits a request to the resource library.

Devices contain many of the unique features of the invention. As an example, arrays are defined as collections of devices which are capable of receiving and producing the desired output. The device hierarchy and built in routing capability allow arrays to break a job down into tasks, one task per copy. The separate tasks are sent to each of the devices constituting the array as each device is ready to receive it. Another example of a unique feature is the intelligent routing accomplished through capabilities and affinities. The logic for routing is built in to each atomic device. The pass/fail response on capabilities and affinity number for a task is passed to the parent device, which then compares the responses from each child device and sends the task to the appropriate device.

Another unique feature of the invention is intelligent translation of a job defined for one type of output device into another. Incoming jobs are often in a data stream that is incompatible with the best fit output device. The intelligent translation device performs the appropriate translation based upon a separate determination of the best fit output device. A current embodiment is capable of translating from POSTSCRIPT to various forms of HP-PCL and PDF.

Other unique features can include color separation where pages with color data are separated from a predominately black and white data stream and sent to a color printer. Most of the document will be printed on a black and white printer which generally has a lower cost per page than color printers. This feature can be implemented by configuring the client print driver to put page boundary markers in the source document data stream.

The resource library and activity log or journal are advantageously coded as separate systems running simultaneously with the main system. The activity journal may be a database containing various tables, entries, queries, and reports relevant to the system. The database interface can be provided by the operating system. The database and its schema, e.g. tables, queries, etc., are created at system startup if they don't already exist. Exemplary database tables might include: an ActionLog which contains system startup and configuration change information; a Billing log which contains originator billing information; a Company log which contains company address information; a FaxList log which relates fax

completion statistics to recipients; a Job log which contains job information, such as start time, stop time, originator, etc.; an Originator log which contains originator information such as address, phone number, etc.; a Recipient log which contains recipient address information; a Recipient list which relates jobs to recipients; a Task log which contains task information such as start time, stop time, production device, etc., and a device log which contains physical device information.

The operator user interface allows the operator to configure the main system for the needs of a particular installation, and is implemented as a separate program from the main system. The main system is capable of operating without the operator user interface running. The operator user interface also saves and loads job templates. Job templates are job tickets that have been saved for later use, and can be edited before submitting a job.

When the user selects "Auto" instead of a specific printer in the graphical user interface, the invention examines the job ticket information to route print jobs to the most effective printer. This feature may be disabled during configuration. If a specific printer is selected by the user, and the printer does not exist, then the job remains unassignable.

Each job is routed to a printer depending on whether the job can be printed at all, printer capabilities, and the best fit of additional performance or post-processing factors, i.e. the affinity of the job to a printer or printer to a job.

Devices have a subset of attributes that define the types of tasks that can be processed. If a task requests a function that is outside the set defined by the device's attributes, then the device is considered to be incapable of processing the task. The attributes include the range of number of pages allowed in a single task, the ability to print color or strictly back and white pages, the ability to print duplex, and the ability to support a requested paper size, color or weight. A task's requirements must fall within all of these restrictions. A task for which no capable devices can be found is considered "Unassignable".

In addition to the above attributes each device is given a unique name, and also has an indicator that specifies if the device should be a candidate during "Automatic Assignment". Automatic Assignment is device selection that is

insensitive to the device's name. If Automatic Assignment is not allowed by a device, and the task does not request that device specifying its name specifically, then the device is considered incapable. If a task requests a specific device, all devices that do not have the name requested are also considered incapable. If no device by the requested name is present in the system, or if no direct path to the requested device is present, then the task is changed to allow Automatic Assignment without regard for originally requested device name. If no device name is ever requested by the task, Automatic Assignment is assumed.

Devices have another subset of attributes that define the device's ability to automate a number of processing options which include the device's processing performance, and the operator's preference toward device. The affinity value for a device is calculated by accumulating the individual affinities given by examining each of the individual attributes.

The automation attributes include the device's ability to collate, to staple, to fold, to drill, to bind, and to add covers. If a task requests one of these functions, the devices that provide the function are given a higher affinity than those devices that do not provide the function. Additional automation functions supported by the device, that are not requested by the task, are simply ignored.

The device's performance is given as a single Impressions Per Minute (IPM) value. The assumption is made that one minute is the optimal average amount of time that a device should spend processing a single task, and that thirty seconds is the standard deviation. A standard bell curve is used to assign relative affinities to each device for a given task.

The operators preference is given as a single value from one to ten. A higher value gives a higher affinity. Each of the above factors is weighted so that a priority relationship between them can be enforced. A higher priority factor will take precedence over any single factor with a lower priority, and the sum of all factors with lower priorities. The priority standings are as follows: 1) Collation; 2) Stapling; 3) Folding; 4) Stitching, Drilling, Binding, and Cover Insertion; 5) Operator Preference; 6) Cost; and 7) Performance

Array Pools and the devices under them have special routing issues. The capability and affinity rules described above must be adjusted to account for these issues. An array is capable of processing a task if any of the devices under it are capable of processing the task. There are two adjustments to the standard capability testing performed by the devices under the array. The allowable number of pages and the requested device name are tested at the array level, not at the subordinate device level. The page range is not used because it is not always known ahead of time how many pages each device in the array will print. The device name testing would allow a maximum of only one device to be capable of defeating the purpose of the array.

The affinity of an array can be determined by averaging all of the affinities of the capable and available subordinate devices. There is only one adjustment to the standard affinity calculations performed by the sub-devices. The device's performance is not factored into the result because, again, the page count for each device is not known. All other affinity factors are evaluated normally.

Figs. 11 through 55 describe a life cycle model of one embodiment of the invention. The life cycle model describes the order in which system operations may occur. The life cycle model, together with the system operation schemata shown specifically in Figs. 13 through 55, fully describe the behavior of the system.

The following rules apply to interpreting the life cycle model and schemata:

Alphabet. Any input or output event may be used in an expression. Output events are prefixed with #.

Operators. Let x and y be life-cycle expression, then:

x.y denotes x is followed by y.

x|y denotes either x or y occurs.

x* denotes zero or more occurrences of x

x~ denotes zero or more occurrences of x simultaneously

[x] denotes that x is optional.

x||y means arbitrarily interleaving the elements of x and y.

Substitutions. An expression can be named in a substitution:

Name = Life-Cycle Expression

Name may be used in other expressions, but substitutions must not be recursive.

Operator precedence. In decreasing order the precedence is:

[ ] , * , + , . , | , | |

Expressions may be bracketed to override default precedence.

The Operation models in the Life Cycle Model are done through textual schemata. Each schema within the schemata shown in Figs. 13 through 55 lists seven sections: (1) Operation - the name of the system operation being described; (2) Description - a free-form abstract of the intent of the operation; (3) Reads - a list of values that are accessed but not changed by the operation; (4) Changes - a list of values that may be modified by the operation; (5) Sends - output events sent by the operation to objects outside the systems (these objects are known as agents); (6) Assumes - a list of conditions that are assumed as being true when the operation begins (if the conditions are not true and the operation is invoked, then the operation's actions and results are undefined); and (7) Result - the conditions and changes in state that are true when the operation has completed.

The recipient and result based paradigms mentioned earlier can be better understood making reference to Figs. 67 - 81. In the recipient based paradigm, a user simply selects the recipient from the recipient list as is shown in Fig. 67. The information is then delivered to that recipient based upon the recipient's preferred device or devices. New recipients can be defined by entering the new recipient's information, such as that shown in Figs. 68 - 70, or possibly as a result of that particular recipient joining the system as a new user by entering new user information, such as that shown in Figs. 71 - 73. Printing and delivery options can